Bisimilarity via unique-solution techniques

Davide Sangiorgi

University of Bologna and INRIA

1 Introduction

Bisimilarity is employed to define behavioural equivalences and reason about them. Finding its origins in concurrency theory, bisimilarity is now widely used also in other areas, as well as outside Computer Science.

In proofs of bisimilarity results, the bisimulation proof method has become predominant, particularly with the enhancements of the method provided by the so called 'up-to techniques' [13]. Among these, one of the most powerful ones is 'up-to context', whereby, during the bisimulation game, a common context in the derivatives of two terms can be erased. Forms of 'bisimulations up-to context' have been shown to be effective in various fields, including process calculi [13, 18, 11], λ-calculi [7, 6, 5, 22], and automata [1, 16]. A lot of work has been carried out recently on 'up-to context' and other up-to techniques, as well as on trying to give a systematic treatment of the set of all such techniques (e.g., the tutorial [13]).

In this paper we summarise a different, ongoing, strand of work about techniques for proving bisimilarity results. This strand is inspired by Milner's theorem about *unique solution of equations* in his landmark book on CCS [8]. This theorem roughly says that two tuples of processes are componentwise bisimilar if they are solutions of the same system of equations. Milner's theorem has however syntactic constraints, on the shape of the equations. For instance, the theorem heavily relies on the sum operator – any other operator but prefix and sum is essentially forbidden. This limits the expressiveness of the technique (since occurrences of other operators above the variables, such as parallel composition and restriction, in general cannot be removed), and its transport onto other languages (e.g., languages for distributed systems or higher-order languages, which usually do not include the sum operator).

An important remark has to be made before providing more technical details. In this paper, behavioural equivalences, hence also bisimilarity, are meant to be *weak* because they abstract from internal moves of terms, as opposed to the *strong* ones, which make no distinctions between the internal moves and the external ones (i.e., the interactions with the environment). Weak equivalences are, practically, the most relevant ones: e.g., two equal programs may produce the same result with different numbers of evaluation steps. The problems mentioned above concerning Milner's theorem do not exist in the case of strong bisimilarity.

We outline two directions for improving Milner's unique-solution technique. The first consists in replacing equations with special inequations called *contractions* [21]. The second consists in replacing the most severe syntactic constraint in Milner's theorem with a semantic condition that has to do with *divergence* [3, 4].

Other motivations for the work on unique solutions are the following. First, the unique-solution techniques convey the flavour of 'bisimilarity up-to context' techinques, as the equations (or contractions) intuitively bring out those contexts that would be erased in an 'up-to context' proof (indeed there are completeness results with respect to certain forms of up-to context, in CCS-like languages [21]). Thus here the goal is to bring light into the up-to-context techniques. For instance, in higher-order languages, while there are well-developed techniques for proving congruence [12], up-to context is still poorly understood [7, 6, 5, 22, 11].

Another possible interest for the unique-solution techniques is that they can be transported onto other equivalences, including contextually-defined equivalences such as barbed congruence, and non-coinductive equivalences such as contextual equivalence (i.e., may testing) and trace equivalence.

2 Equations and Milner's theorem

We recall equations and Milner's theorem, in the setting of CCS. We omit the standard definitions of syntax and operational semantics of the calculus, as well of weak bisimulation, written $\approx$, [8].

Uniqueness of solutions of equations [8] intuitively says that if a context C obeys certain conditions, then all processes P that satisfy the equation $P \approx C[P]$ are bisimilar with each other. We use capital letters X, Y, Z for variable of equations. The body of an equation is a CCS expression possibly containing variables.

Definition 1. *Given, for each i of a countable indexing set I, variables X_i, and expressions E_i possibly containing such variables, $\{X_i = E_i\}_{i \in I}$ is a* system of equations. *(There is one equation for each variable X_i.)*

We write $E[\widetilde{P}]$ for the expression resulting from E by replacing each variable X_i with the process P_i

Definition 2. *Suppose $\{X_i = E_i\}_{i \in I}$ is a system of equations:*

- *$\widetilde{P}$ is a* solution of the system of equations for $\approx$ *if for each i it holds that $P_i \approx E_i[\widetilde{P}]$.*
- *the system has* a unique solution for $\approx$ *if whenever $\widetilde{P}$ and $\widetilde{Q}$ are both solutions for $\approx$, then $\widetilde{P} \approx \widetilde{Q}$.*

Definition 3. *A system of equations $\{X_i = E_i\}_{i \in I}$ is*

- guarded *if, in each E_i, each occurrence of an equation variable is underneath a visible prefix (i.e., a prefix that is not τ);*
- sequential *if, in each E_i, each occurrence of an equation variable only appears underneath prefixes and sums.*

In other words, if the system is sequential, then for every expression E_i, any subexpression of E_i in which X_j appears, apart from X_j itself, is a sum (of prefixed terms). For instance, $X = \tau.X + \mu.\mathbf{0}$ is sequential but not guarded; using ℓ for a visible prefix, $X = \ell.X \mid P$ is guarded but not sequential, whereas $X = \ell.X + \tau.\boldsymbol{\nu} a\,(a.\overline{b} \mid a.\mathbf{0})$, as well as $X = \tau.(a.X + \tau.b.X + \tau)$ are both guarded and sequential.

Theorem 1 (unique solution of equations, [8]). *A system of guarded and sequential equations has a unique solution for* $\approx$. □

3 Contractions

Intuitively, for a behavioural equivalence $\asymp$, its contraction $\succeq_{\asymp}$ is a preorder in which $P \succeq_{\asymp} Q$ holds if $P \asymp Q$ and, in addition, Q has the *possibility* of being at least as efficient as P. That is, if P can do some work (i.e., some interactions with its environment), then Q should be able to do the same work at least as quickly as P (i.e., performing no more τ-steps then those performed by P). Process Q, however, may be nondeterministic and may have other ways of doing the same work, and these could be slow (i.e., involving more τ-steps than those performed by P). The *bisimilarity contraction* is written $\succeq_{\mathrm{bis}}$ and is a precongruence in CCS; see [20, 21] for the formal definition. A *system of contractions* is defined as a system of equations, except that the contraction symbol $\succeq$ is used in the place of the equality symbol $=$. Thus a system of contractions is a set $\{X_i \succeq E_i\}_{i\in I}$ where I is an indexing set and expressions E_i may contain the contraction variables $\{X_i\}_{i\in I}$.

Definition 4. *Given a behavioural equivalence* $\asymp$ *and its contraction* $\succeq_{\asymp}$*, and a system of contractions* $\{X_i \succeq E_i\}_{i\in I}$*, we say that:*

- $\widetilde{P}$ *is a* solution for $\succeq_{\asymp}$ of the system of contractions *if* $\widetilde{P} \succeq_{\asymp} \widetilde{E}[\widetilde{P}]$*;*
- *the system has* a unique solution for $\asymp$ *if whenever* $\widetilde{P}$ *and* $\widetilde{Q}$ *are both solutions for* $\succeq_{\asymp}$ *then* $\widetilde{P} \asymp \widetilde{Q}$*.*

When we reason about bisimilarity, the contraction symbol $\succeq$ is interpreted as the bisimilarity contraction $\succeq_{\mathrm{bis}}$, and the equivalence $\asymp$ as the bisimilarity $\approx$. Thus $\widetilde{P}$ being a solution for $\succeq_{\mathrm{bis}}$ of the system of contractions $\{X_i \succeq E_i\}_{i\in I}$ means that $\widetilde{P} \succeq_{\mathrm{bis}} \widetilde{E}[\widetilde{P}]$; and the system having a unique solution for $\approx$ means that whenever $\widetilde{P}$ and $\widetilde{Q}$ are both solutions for $\succeq_{\mathrm{bis}}$ then $\widetilde{P} \approx \widetilde{Q}$.

Lemma 1. *If a system of equations* $\{X_i = E_i\}_{i\in I}$ *has a unique solution for* $\approx$*, then also the corresponding system of contractions* $\{X_i \succeq E_i\}_{i\in I}$ *has a unique solution for* $\approx$*.*

The converse of the lemma, in contrast, is false: systems of contractions more easily have a unique solution.

Definition 5. *A system of contractions* $\{X_i \succeq E_i\}_{i\in I}$ *is* weakly guarded *if, in each* E_i*, each occurrence of a contraction variable is underneath a prefix.*

Theorem 2 (unique solution of contractions for $\approx$). *A system of weakly-guarded contractions has a unique solution for* $\approx$*.*

We refer to [20, 21] for discussions on completeness (the method is complete, in the sense that any process bisimilarity can be proved with the method), examples of application, abstract formulation of the method, and generalisation to other languages and to other equivalences, including non-coinductive equivalences.

4 Divergence in equations

In this section we highlight a different approach, whereby one goes back to equations, but adds a condition based in divergence. In its basic form, the method (inspired by results by Roscoe in CSP [15, 14] essentially says that a guarded equation (or system of equations) whose infinite unfolding never produces a divergence has the unique-solution property.

We discuss the approach in CCS, as before, and assuming for simplicity a single equation $X = E$ (the results can be generalised to systems of equations). We need to reason with the unfoldings of the given equation $X = E$: we define the n-th unfolding of E to be E^n; thus E^1 is defined as E, E^2 as $E[E]$, and E^{n+1} as $E^n[E]$. The *infinite* unfolding represents the simplest and most intuitive solution to the equation. In the CCS grammar, such a solution is obtained by turning the equation into a recursive definition, namely the process K_E with $K_E \stackrel{\triangle}{=} E[K_E]$. Process K_E is the *syntactic solution of the equation.*

Theorem 3 (Unique solution). *A guarded system of equations whose syntactic solutions do not diverge has a unique solution for* $\approx$.

The theorem can be strengthened by distinguishing between different forms of divergence; in particular, ignoring divergences that already show up in partial unfoldings of the equations, i.e., E^n for some $n \geq 0$, called *innocuous* divergences.

We refer to [3] for abstract formulations of the method, application to other languages, equivalences, as well as preorders. In [4] the method is applied to showing what is the equivalence induced on λ-terms by Milner's encoding into the π-calculus (for call-by-value), a problem that had remained open since Milner's work on functions as processes [10, 9]. Such proofs seem hard to carry out with the ordinary bisimulation proof method, even when enhanced by means of 'up-to techniques'.

4.1 Comparison

In comparison with the method based on contractions, the main drawback for the method based on equations is the presence of a semantic condition, involving divergence: the unfoldings of the equations should not produce divergences, or only produce innocuous divergences. Various techniques for checking divergence exist in the literature, including type-based techniques [23, 19, 2]; a syntactic condition is proposed in [3]. However, in general divergence is undecidable, and therefore, the check may sometimes be unfeasible. Nevertheless, the equations that one writes for proofs usually involve forms of 'normalised' processes, and as such they are divergence free (or at most, contain only innocuous divergences). More experiments are needed to validate this claim or to understand how limiting this problem is.

On the other hand, using contractions for proving an equivalence, one needs also the theory of the associated contraction preorder; moreover there may be processes for which the contraction technique is not applicable simply because the contraction preorder is strictly finer than the equivalence, and therefore one of the processes fails to be a solution.

References

1. Bonchi, F., Pous, D.: Checking nfa equivalence with bisimulations up to congruence. In: Giacobazzi, R., Cousot, R. (eds.) Proc. POPL'13. pp. 457–468. ACM (2013)
2. Demangeon, R., Hirschkoff, D., Sangiorgi, D.: Termination in impure concurrent languages. In: Proc. 21th CONCUR. LNCS 6269, pp. 328–342. Springer (2010)
3. Durier, A., Hirschkoff, D., Sangiorgi, D.: Divergence and unique solution of equations. In: 28th CONCUR. LIPIcs, vol. 85, pp. 11:1–11:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017)
4. Durier, A., Hirschkoff, D., Sangiorgi, D.: Eager functions as processes. In: 33nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018. IEEE Computer Society (2018)
5. Koutavas, V., Wand, M.: Small bisimulations for reasoning about higher-order imperative programs. In: Proc. 33rd POPL. pp. 141–152 (2006)
6. Lassen, S.: Relational reasoning about contexts. In: Higher-order operational techniques in semantics. pp. 91–135. Cambridge University Press (1998)
7. Lassen, S.: Bisimulation in untyped lambda calculus: Böhm trees and bisimulation up to context. Electr. Notes Theor. Comput. Sci. 20, 346–374 (1999)
8. Milner, R.: Communication and Concurrency. Prentice Hall (1989)
9. Milner, R.: Functions as processes. Journal of Mathematical Structures in Computer Science 2(2), 119–141 (1992)
10. Milner, R.: Functions as processes. In: ICALP. LNCS 443, pp. 167–180. Springer (1990)
11. Piérard, A., Sumii, E.: Sound bisimulations for higher-order distributed process calculus. In: Proc. FOSSACS. LNCS 6604, pp. 123–137. Springer (2011)
12. Pitts, A.: Howe's method. In: Sangiorgi, D., Rutten, J. (eds.) Advanced Topics in Bisimulation and Coinduction. Cambridge University Press (2012)
13. Pous, D., Sangiorgi, D.: Enhancements of the bisimulation proof method. In: Sangiorgi, D., Rutten, J. (eds.) Advanced Topics in Bisimulation and Coinduction. Cambridge University Press (2012)
14. Roscoe, A.W.: An alternative order for the failures model. J. Log. Comput. 2(5), 557–577 (1992)
15. Roscoe, A.W.: The theory and practice of concurrency. Prentice Hall (1998).
16. Rot, J., Bonsangue, M.M., Rutten, J.J.M.M.: Coinductive proof techniques for language equivalence. In: Proc. LATA. LNCS 7810, pp. 480–492. Springer (2013)
17. Sangiorgi, D., Milner, R.: The problem of "Weak Bisimulation up to". In: Proc. CONCUR '92. LNCS 630, pp. 32–46. Springer Verlag (1992)
18. Sangiorgi, D., Walker, D.: The π-calculus: a Theory of Mobile Processes. Cambridge University Press (2001)
19. Sangiorgi, D.: Termination of processes. Mathematical Structures in Computer Science 16(1), 1–39 (2006)
20. Sangiorgi, D.: Equations, contractions, and unique solutions. In: POPL 2015. pp. 421–432. ACM (2015),
21. Sangiorgi, D.: Equations, contractions, and unique solutions. ACM Trans. Comput. Log. 18(1), 4:1–4:30 (2017),
22. Sangiorgi, D., Kobayashi, N., Sumii, E.: Environmental bisimulations for higher-order languages. ACM Trans. Program. Lang. Syst. 33(1), 5 (2011)
23. Yoshida, N., Berger, M., Honda, K.: Strong Normalisation in the Pi-Calculus. Information and Computation 191(2), 145–202 (2004)